SOFTWARE ARTS TECHNICAL NOTES, BY THE CREATORS OF VISICALC

A Publication of
Software Arts, Inc.

SATN

THE JOURNAL FOR VISICALC USERS

Vol. I, No. 2
November/December 1981

IN THIS ISSUE:

Mortgages and Loans 1
  Calculating the Monthly Payment
  Calculating the Annual Interest Costs
  After Tax Costs
  Extending the Sheet
The Saved Sheet 5
  The Save and Load Commands
  Printing out the Expressions
Carrying on With VisiCalc 8
  Building a Cryptarithm on a VisiCalc Sheet
  Adding the Cryptarithm
  It's What's Left That Counts
  Creating Mod with @INT
  Carry On
  Summing Up
Who's in First? 11
  Baseball Standings
  Hockey and Football Standings
. . . Plus VisiCalc Pointers

MORTGAGES AND LOANS

When you cannot, or do not want to, pay cash for a big ticket item like a boat, a car, or a house, then the question "What's the monthly payment?" is one of the first you are likely to ask. The answer depends, of course, on the price of the thing you are buying, the amount of your downpayment, the interest rate charged by the lender, and the time you can take to pay off the debt.

Given those elements, one common way to answer the questions is to consult a repayment table. However, such tables are often organized so that if you wish to find the payment required to repay, or amortize, a loan of, say, $46,500, you have to sum the payments necessary to amortize loans of $40,000, $6,000, and $500. And to save space and cost, the publishers of such tables may use interest rate increments that exclude the exact rate you are seeking.

Mortgage amortization tables are even less helpful if you want to know how much interest you will pay each year. Since the interest on a loan is usually tax deductible, a loan that appears beyond your financial reach may actually be within it when the interest deduction is calculated (assuming that you can carry the loan until the tax deduction takes effect).

Using the VisiCalc® program can greatly simplify the task of calculating monthly payments. It can also be used to calculate each year's interest deductions, allowing you to evaluate the true cost of the loan, and map out tax strategies more efficiently. In short, it can make shopping for money easier.

The example we will concentrate on in this article is a mortgage, a fixed rate, long term loan. The formulas and techniques described, however, are useful for any kind of purchase for which credit is extended, at any fixed interest rate, and for any repayment period. Our example is based on a monthly repayment schedule, but with minor alterations can also be used for quarterly, semiannual, or other repayment schedules.

Calculating the Monthly Payment

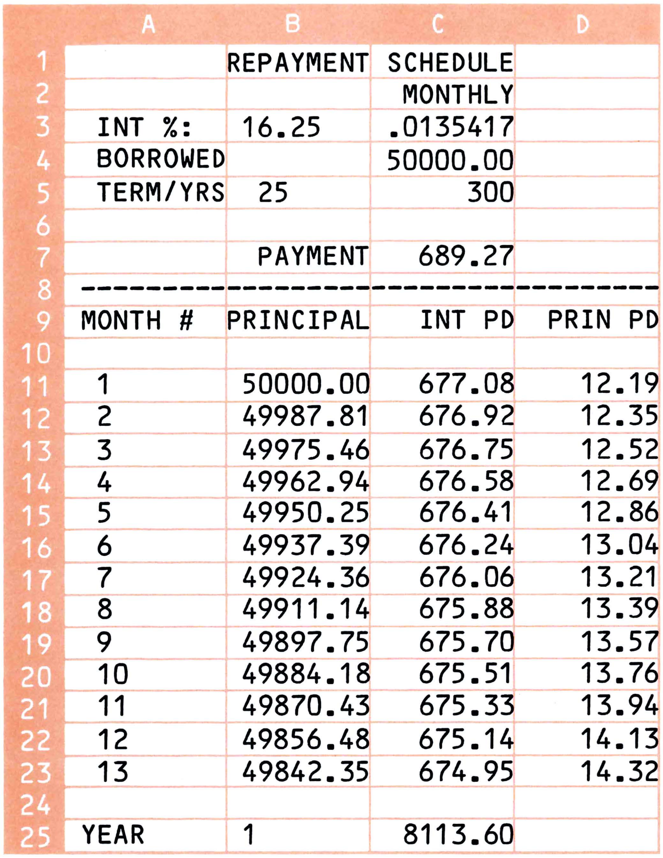
The sheet shown in Figure 1 is set up with the commands /GOR, to calculate across the rows, and /GRM, to allow us to change such variables as the interest rate, the term of the loan, and the amount borrowed before recalculating with !. We have also formatted the sheet with the command /GF$.

The interest rate is entered with the %INTEREST row at B3 as a percentage figure, in our example as 16.25. The monthly rate of interest is at C3 and is the stated rate divided by 100 and then divided by 12. Note that this cell is formatted with /FG to show the precise monthly rate. Enter:

```
>C3:/FG+B3/100/12
```

To calculate the interest rate for any other period, substitute the number of payments per year for 12 in the formula entered.

The mortgage amount is entered on the BORROWED row at C4, and the term of the mortgage (in



	A	B	C	D
1		REPAYMENT	SCHEDULE	
2			MONTHLY	
3	INT %:	16.25	.0135417	
4	BORROWED		50000.00	
5	TERM/YRS	25	300	
6				
7		PAYMENT	689.27	
8	---------	---------	---------	---------
9	MONTH #	PRINCIPAL	INT PD	PRIN PD
10				
11	1	50000.00	677.08	12.19
12	2	49987.81	676.92	12.35
13	3	49975.46	676.75	12.52
14	4	49962.94	676.58	12.69
15	5	49950.25	676.41	12.86
16	6	49937.39	676.24	13.04
17	7	49924.36	676.06	13.21
18	8	49911.14	675.88	13.39
19	9	49897.75	675.70	13.57
20	10	49884.18	675.51	13.76
21	11	49870.43	675.33	13.94
22	12	49856.48	675.14	14.13
23	13	49842.35	674.95	14.32
24				
25	YEAR	1	8113.60	

FIGURE 1. *Calculations of the monthly payment and annual interest charges on a $50,000 mortgage.*

years) at B5. At C5 the number of monthly payments is calculated by the formula:

```
>C5:+B5*12
```

Once again, substitute for 12 the appropriate number for any other payment schedule.

The general formula to calculate the level payment required to amortize a loan is:

$$\frac{Interest}{1-(1+Interest)^{-Term}} * Sum\ Borrowed$$

Where *Interest* is expressed as a decimal, and *Term* as the number of payments.

On our sheet the formula is expressed by entering:

```
>C7:+C3/(1-((1+C3)^-C5))*C4
```

After you've entered that expression on the sheet, you can calculate the payment on any loan with a fixed rate of interest, for any period.

Calculating the Annual Interest Costs

Once the level monthly payment is determined, we can create a table showing the annual interest payments. The first and only entered value in the table is the original amount borrowed. All remaining values are calculated by simple formulas replicated down the sheet.

To create the table, the amount of the mortgage is entered at B11 as:

```
>B11:+C4
```

At C11 we calculate the amount of interest on the unpaid principal by multiplying it by the interest rate for one month. Enter:

```
>C11:+C3*B11
```

That produces the amount of interest paid for the first month of the mortgage. In our example it shows that $677.08 of the $689.27 is interest, meaning that in the next month $12.19 is deducted from the unpaid principal against which the monthly interest rate is to be applied. To calculate how much principal is paid, then, we subtract the amount of interest paid in the first month from the monthly payment. Enter:

```
>D11:+C7-C11
```

The next step is to calculate the new amount of unpaid principal, which is a matter of deducting the result of the last calculation from the first amount of principal. Thus at B12, enter:

```
>B12:+B11-D11
```

Once those three formulas are entered on the sheet, they can be replicated down the sheet. However, if you try to replicate them too far, you'll begin running low on memory. An easy way around that is simply to replicate the formulas for the table down to thirteen months.

```
>B12:/R:B13...B23:RR
>C11:/R:C12...C23:NR
>D11:/R:D12...D23:NR
```

Then, to calculate the first year's interest payments, use the @SUM function for the first twelve months. Thus at C25, enter:

```
>C25:@SUM(C11...C22)
```

To keep track of which year's interest has been calculated we include at B25 the formula:

```
>B25:+A22/12
```

SATN is published bi-monthly by Software Arts, Inc., P.O. Box 494, Cambridge, Massachusetts 02139. Address all editorial correspondence to the Editorial Department, SATN, P.O. Box 494, Cambridge, MA 02139. Requests for special permission or bulk orders should be addressed to the publisher.

Domestic subscriptions: 6 issues, $30.00; outside U.S.A., $50.00. Single copy price, $6.00. Send subscriptions, change of address, and fulfillment questions to SATN Subscriptions, P.O. Box 815, Quincy, MA 02169.

Copyright © 1981, Software Arts, Inc. All rights reserved. This periodical may not be copied or reproduced in any fashion either in whole or in part without the prior written consent of Software Arts, Inc.

Software Arts, Inc. makes no warranty, express or implied, with respect to the quality, accuracy or freedom from error of the information contained in SATN, the journal for VisiCalc users, and specifically disclaims all liability for any damages resulting from the use of such information.

Apple is a registered trademark of Apple Computer Inc. Atari is a trademark of Atari Inc. Commodore is a trademark of Commodore International. DIF is a trademark of Software Arts, Inc. HP-83/85 and HP-125 are trademarks of Hewlett-Packard Company. IBM is a registered trademark of International Business Machines Corporation. TRS-80 is a trademark of Radio Shack Division of Tandy Corp.

SATN is a trademark of Software Arts, Inc.
VisiCalc is a registered trademark of Personal Software Inc.

The thirteenth month is not included in the calculations for the year, but is included in the table so that we can recalculate each succeeding year's interest with !. To see how that works, enter the following:

```
>A11:+A23
>B11:+B23®!
```

That replaces month 1 with month 13, and the amount of unpaid principal with $49842.35, the amount that would be unpaid at the beginning of the second year. When ! is used, the columns are recalculated to produce a table for the second year. Each additional use of ! will produce another year's table and the sum of the interest paid that year.

Before saving your sheet, however, you should return the cursor to A11 and reenter 1, and to B11 and reenter +C4. If you don't, then when you load the sheet the table may consist entirely of ERRORs. That's because when a VisiCalc sheet is loaded, all calculated values may be set to ERROR. A recalculation, performed by the program before the sheet is displayed, sets the calculated values to the correct amounts. However, if the values in row 11 contain references to values in row 23, that is, forward references, when the recalculation is performed on load, the program finds ERROR in row 23, returns it to row 11, and it is propagated throughout the table. Should you not reenter 1 and +C4 before you save the sheet, however, you can do so after the sheet is loaded, and use ! to restore the table.

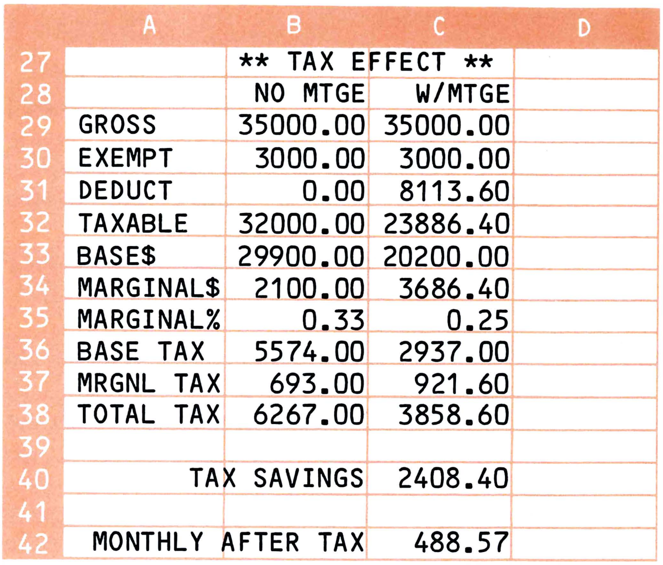
After Tax Costs

Once the annual interest has been determined, you can calculate the true cost of carrying a mortgage by calculating the effect of the tax subsidy the IRS provides to you.

To show how much an interest deduction is worth we can look at an example. Imagine that a married couple has one child and an income of $35,000. They file a joint return claiming three exemptions and do not own a home, but rent an apartment. They have too few deductions to justify itemizing, but buying a house or condominium generates interest payments that make it sensible to itemize. By itemizing, the couple can reduce their tax burden by deducting the mortgage interest as well as property taxes, state income taxes, medical expenses, charitable contributions, and so on. For simplicity's sake, however, our example shows only the effect of the mortgage interest deduction.

Figure 2 is a continuation of the sheet shown in Figure 1. Not shown is an @LOOKUP table that we have entered on the sheet in another location. The @LOOKUP table, shown in Figure 3, is taken from the IRS income tax table for married persons filing jointly. Only the ranges and corresponding values differ from the @LOOKUP table shown in the September/October issue of SATN, Vol. I, No. 1. Its construction is identical.

The lines GROSS, for gross income, and EXEMPT, for the number of exemptions claimed, contain values that vary from case to case. Since the couple claims three exemptions, each of which is worth $1000, EXEMPT is shown at B30 as 3000.00. The values for



	A	B	C	D
27		** TAX EFFECT **		
28		NO MTGE	W/MTGE	
29	GROSS	35000.00	35000.00	
30	EXEMPT	3000.00	3000.00	
31	DEDUCT	0.00	8113.60	
32	TAXABLE	32000.00	23886.40	
33	BASE$	29900.00	20200.00	
34	MARGINAL$	2100.00	3686.40	
35	MARGINAL%	0.33	0.25	
36	BASE TAX	5574.00	2937.00	
37	MRGNL TAX	693.00	921.60	
38	TOTAL TAX	6267.00	3858.60	
39				
40		TAX SAVINGS	2408.40	
41				
42	MONTHLY AFTER TAX		488.57	

FIGURE 2. *The calculation of the tax effect of a mortgage deduction. Note that GROSS can be either your gross income or, if you have any loss or gains to report, your "adjusted gross income."*

KEY TO CONVENTIONS USED IN SATN

The VisiCalc program is available on personal computers manufactured by Apple, Atari, Commodore, Hewlett-Packard, IBM, and Radio Shack. To alleviate possible confusion arising from the differences among the keyboards of these computer systems, SATN will use a standardized symbol and name to represent certain commonly used keys.

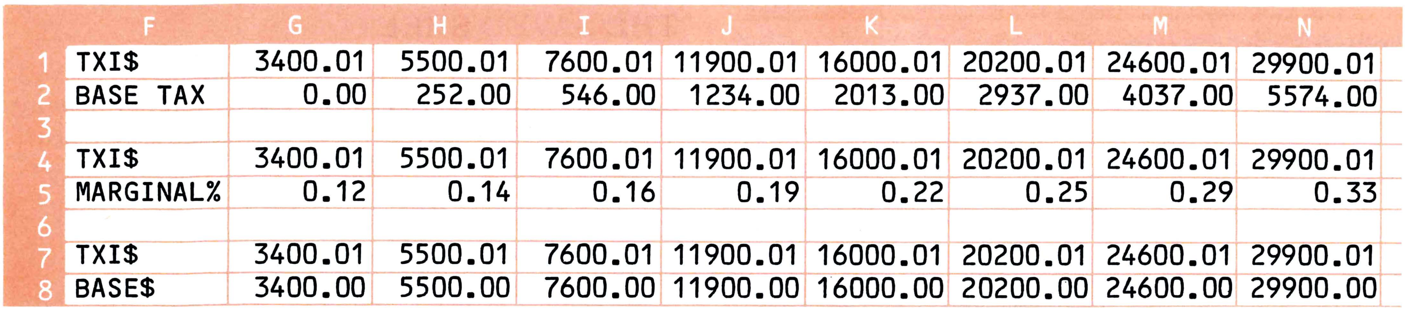
In this issue, the symbol ® and the word RETURN are used synonymously to represent the keys known as Enter, Endline, and Return. The symbol ∧ is used to represent an exponentiation operator.

Also, the VisiCalc program was designed so that the colon (:) has the same effect as RETURN in some commands, such as to end the > command and to separate the various elements of the replicate command. The reason for this is to aid in making the documentation more readable.

In examples of what to type, type exactly what you see and press RETURN at the end of every line. Throughout the journal, SATN will use one particular typeface to represent both VisiCalc input and output (what you should type and what you will see on the screen). The typeface looks like this:

```
This line is typeset in OCR-B.
 So is this: +A1+B1*3.25
```

To allow you to type the formulas directly into VisiCalc, the examples use a specific coordinate such as A1 rather than using the name of the value it represents. When actually typing in a formula, you can of course use the arrows to point instead of typing in the coordinate name.



	F	G	H	I	J	K	L	M	N
1	TXI$	3400.01	5500.01	7600.01	11900.01	16000.01	20200.01	24600.01	29900.01
2	BASE TAX	0.00	252.00	546.00	1234.00	2013.00	2937.00	4037.00	5574.00
3									
4	TXI$	3400.01	5500.01	7600.01	11900.01	16000.01	20200.01	24600.01	29900.01
5	MARGINAL%	0.12	0.14	0.16	0.19	0.22	0.25	0.29	0.33
6									
7	TXI$	3400.01	5500.01	7600.01	11900.01	16000.01	20200.01	24600.01	29900.01
8	BASE$	3400.00	5500.00	7600.00	11900.00	16000.00	20200.00	24600.00	29900.00

FIGURE 3. *@LOOKUP table for married taxpayers filing joint returns. TXI$ stands for Taxable Income.*

GROSS and EXEMPT are entered in column C with the entries:

```
>C29:+B29
>C30:+B30
```

The third line, DEDUCT, contains the amount of itemized deductions available to the taxpayer. Because the couple in our example does not itemize until they assume a mortgage, and because we will show only the effect of the mortgage deduction, we have entered 0.00 at B31, and + C25, which is the amount of interest paid on the mortgage, at C31. That way, the amount of interest paid in any year is substituted after recalculating with !.

TAXABLE is simply the subtraction of EXEMPT and DEDUCT from GROSS and is calculated at B32 and C32 by entering:

```
>B32:+B29-B30-B31
>B32:/R:C32:RRR
```

BASE$ is the amount of taxable income shown in the IRS table against which a fixed dollar amount of tax is levied. It is entered at B33 and C33 by using the @LOOKUP table. Thus, enter:

```
>B33:@LOOKUP(B32,G7 . . . R7)
>B33:/R:C33:RNN
```

MARGINAL$ is taxable income less BASE$, and is that portion of taxable income against which the taxpayer's so-called marginal tax rate is applied. It is calculated at B34 and C34 by entering:

```
>B34:+B32-B33
>B34:/R:C34:RR
```

MARGINAL% is the marginal tax rate, which, like BASE$, comes from the @LOOKUP table. It is found by entering:

```
>B35:@LOOKUP(B32,G4 . . . R4)
>B35:/R:C35:RNN
```

BASE TAX is the amount of tax that is levied on BASE$. It, too, is taken from the @LOOKUP table, and is entered for both columns by:

```
>B36:@LOOKUP(B32,G1 . . . R1)
>B36:/R:C36:RNN
```

That yields the basic amount of tax to be paid, to which we must add the marginal tax, which is the result of multiplying MARGINAL$ by MARGINAL%. Enter:

```
>B37:+B35*B34
>B37:/R:C37:RR
```

Now, all that remains to find TOTAL TAX is to add BASE TAX and MRGNL TAX, which is done by entering:

```
>B38:+B36+B37
>B38:/R:C38:RR
```

TAX SAVINGS is the calculation of how much the deduction is worth in dollars and cents, that is, the amount of the mortgage interest paid by the IRS, and is the result of deducting the tax the couple would pay if they have a mortgage from the tax they would pay if they had none. It's found by entering:

```
>C40:B38-C38
```

Another way of looking at the same figure is in terms of its effect on the monthly payment. To calculate the monthly PAYMENT AFTER TAX, we divide TAX SAVINGS by 12 and deduct the result from the monthly payment, as follows:

```
>C42:+C7-(C40/12)
```

Extending the Sheet

It's possible to create a sheet even more extensive than this one by adding other calculations and expressions at different points. For example, under LOAN REPAYMENT you may wish to add rows to include the actual purchase price of the house, the amount of your downpayment, closing costs, title insurance, points, and so forth. Or you could calculate the present value of your monthly payments with @NPV.

If you are willing to be reminded, you can multiply the number of payments by the payment, add the downpayment, and find out how much you will actually be paying for the house. And once you've calculated that, by subtracting the amount borrowed from the amount repaid, you can learn how much you'll pay in interest.

In the TAX EFFECT portion of the sheet, of course, you may insert lines that would enable you to include, item by item, those expenses you are entitled to deduct. You may also want to replicate the W/MTGE column into a few other columns, change GROSS, and examine the differing effects your itemized deductions have at differing income levels.

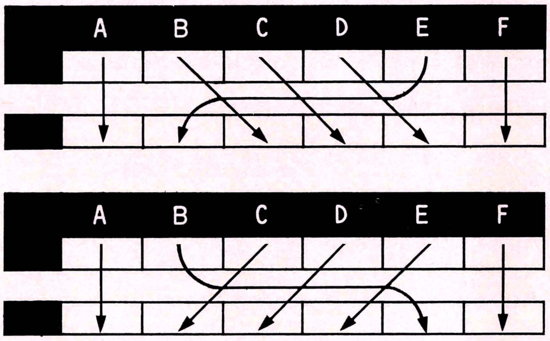
In short, the sheet we've developed, while useful as it stands, can be readily expanded upon and altered to fit your specific situation.

VISICALC POINTER . . .

The /M command seems to operate differently when you are moving a row up rather than down or a column to the left rather than the right.

In many word processing programs, material that is being inserted in the text always appears directly to the left of, or directly above, the cursor. The /M command operates in the same way as those programs. That is, the column or row being moved will always appear directly to the left of, or directly above, the cursor's position at the time the command is terminated. The /M command operates consistently whether you are moving a column to the left or right, or a row up or down. That's to say that no matter which direction you're moving a row or column, relative to the cursor its final location will always be the same.

Much of the confusion concerning the operation of /M has arisen because of the contents of the prompt line. There is a foolproof way to read the prompt line so that the row or column you are moving invariably ends where you want it. To the prompt MOVE: FROM . . . TO, simply add the words "the position above (or to the left of) the row (or column) *currently at* . . . " and supply the coordinate specified on the edit line. In that sense, then, the /M command operates the same way as the insert command. When you use /IR or /IC, the new row or column appears directly above or to the left of the cursor.



As shown in these diagrams, when moving column E to the position shown, the prompt and edit lines will read "MOVE:FROM . . . TO" and "E1 . . . B1". Column E will then move *to the left of the position currently occupied by B1*. When moving column B to the right, the prompt and edit lines will read "MOVE:FROM . . . TO" and "B1 . . . F1". Column B will then move *to the left of the position currently occupied by F1.*

After using the /M command for a while, you will probably find that you no longer refer to the prompt line, but point with the cursor to the place where you want the source row or column to appear. When you are moving a column to the right or left, move the cursor to the column immediately to the right of the target column and terminate the /M command with a ®.

When you move a row up or down, position the cursor immediately beneath the row to which you wish to move the source row and terminate the command with a ®. As with a column move, all the rows between the source row and the target row will shift one space.

THE SAVED SHEET

Each time you suspend the development of a VisiCalc® sheet to turn your attention to another activity, you use /SS to store your work. The purpose of this article is to explore several aspects of the /SS and /SL commands.

The Save and Load Commands

One of the first things to understand about the method by which a VisiCalc sheet is saved is the kind of file that the program writes. Specifically, it is an ASCII text file. A similar ASCII text file on another machine would load compatibly. We at Software Arts, Inc. chose that format because such files are easily read by BASIC programs as well as other programs in other languages. That means VisiCalc files can be read by many other programs, including some commercially available word processing programs and programs written by users.

Entering the /SS command causes the VisiCalc program to save a sheet in the form of keystrokes (though not necessarily the original keystrokes used) that would recreate the sheet if they were typed in directly. Then, when you use /SL to load a saved sheet, the VisiCalc program notes for itself that it is reading from a disk and treats the data and organization of the file almost as though it were being typed from the keyboard.

The VisiCalc program does not perform calculations, or evaluate expressions, while the sheet is loading. Rather, when the reading of the file has been completed, the last step before the sheet is displayed is a single recalculation of the entire sheet. As the sheet loads, the values of all expressions may be set to ERROR. Consequently, if there is a forward reference on the sheet, the recalculation performed before the sheet is displayed will cause all values dependent on that entry position to appear as ERROR when the sheet appears on the screen.

When a sheet is saved, only certain commands are written to the file. There is no guarantee that future versions of the VisiCalc program will continue to save all the same commands, nor save expressions in the same order that it currently saves them. However, you will continue to be able to load sheets saved by old VisiCalc programs in new versions of VisiCalc. Also, not all commands that the VisiCalc program accepts from the keyboard will work when read from the disk.

Printing Out the Expressions

There are times when it is easier to edit and audit the expressions on a hard copy than on the screen, and having a printout of the expressions becomes useful. Further, the information you store on a data disk can be lost, as you may have learned if you ever subjected one to a charge of static electricity, or you might simply lose the disk itself. In any event, having a printed record of the formulas and format commands on your VisiCalc sheet allows you to reproduce the sheet, exactly as it appeared when you gave the /SS command, by typing the commands from the printout.

TABLE 1

Since printing out the formulas and formatting information on your sheet is essentially a process of saving to a printer, rather than to a disk, the general command used is /SS. That portion of the full command required is common to all the personal computers that run the VisiCalc program. Thereafter, because of the differences between the operating systems of the machines, the command differs slightly from system to system. Note that in the table below, the letter *n* identifies the slot or port number.

COMPUTER	FULL COMMAND	REMARKS
Apple II*	/SS,SP	Use this command and the Apple will find the printer slot.
	/SS,S*n*	Use this command when the printer card location is known.
Atari	/SSP:	
Commodore	/SS,TP /SS,TA,CA	Use with Commodore printers. Use with ASCII printers.
HP 83/85	/SS	Print the formulas with the VZREAD program.
HP 125	/SSINT: /SSHPB:	Use with the internal printer. Use with the Hewlett-Packard printer. (Other printers require different specifications. See the manual.)
IBM	/SSLPT*n*:	Use with line printers.
	/SSCOM*n*:	Use with serial printers.
TRS80 (I, II, III)	/SS:P	

* Version 1.37 of the VisiCalc program does not print directly, but you may copy the disk file to the printer.

The formulas and formats on your sheet are printed with /SS and a device specification that depends on the computer you are using. Table 1 shows the appropriate commands for the different computers that the VisiCalc program runs on. Regardless of which system you use, there are only minor differences in the printouts that are generated when you enter the appropriate commands in Table 1.

One of the first things you notice about the printouts is that they are in reverse order. That is, the first entry in the printout is that of the cell in the lower right corner of the sheet, the farthest cell from the beginning of the sheet. We designed the program that way so that the sheet being loaded would only have to be "grown" once, and so would require fewer time consuming memory reallocations.

Each line specifying the contents of a cell begins with sufficient information to get you to the right location on the sheet; that is, a "go to coordinate" (>) command, followed by the cell's coordinate, followed by a colon. The colon, as used here, has the same effect as typing ®. We chose to make it work that way to make the printouts more readable. If you type >A1:, the cursor will move to cell A1, and you can then type in the rest of the information from the line of the printout, terminating the instruction with a ®.

Look at the printout in Figure 4, and compare it with Figure 5, a screen photo of the sheet. In the screen photo, the two windows begin respectively with the labels "LEFT" and "RIGHT," which is how we will refer to them hereafter.

We'll proceed through the printout in Figure 4, from top to bottom, examining the lines highlighted in rust. As you'll see, every line on the printout is recorded in such a way that it can be typed at the keyboard.

At S23 on the printout is the line:

```
>S23:+S22+2
```

Enter that line from the keyboard and the cursor will go to S23 when you type the :, and then will cause the program to add 2 to whatever is in S22.

The line:

```
>R22:/FR"A
```

formats cell R22 with /FR, and then, as indicated by the ", enter a label consisting of the letter A. The " can be used to create any label, and must be used to create any label that begins with a character that is not alphabetic.

The line:

```
>R21:/-*-
```

indicates that at R21 there is a repeating label (/–) consisting of as many repetitions of * – as are required to fill the width of the column.

At D14 the screen photo shows nothing, but the printout shows an entry; the format command /FL.

The VisiCalc program writes such information to the file, even though the screen displays nothing. In other words, you could fill an entire sheet with local format commands, have what appears to be an empty sheet, and still have information that would be saved if you used /SS. Keep in mind, as you replicate local format commands, that they may use more space on disk than they do in memory. If you use a lot of local format commands, especially in blank locations, you will use up a lot of disk space. If you find that your files are too big, you might check to see if you have crowded them with local format commands in otherwise empty cells. A global format command consumes very little disk space and may serve your purpose as well.

Following the line:

```
>A11:"LEFT
```

the printout lists a series of commands that set up the screen for display. The first three, /W1, /GOC, and /GRA are familiar ones, and set the window to 1, the order of recalculation to column, and the recalculation mode to automatic.

The next command, beginning with /X, is an unfamiliar one to many VisiCalc users. Though they appear on printouts, /X commands are purposely not documented or supported. They are used by the VisiCalc program to simplify the process of recreating the sheet, and the details of their actual behavior vary from machine to machine and from one version of the VisiCalc program to another. The net effect of their operation is to reproduce the sheet exactly as it appeared when it was saved.

```
>T29:"THERE
>S23:+S22+2
>R23:/FR"B
>S22:2
>R22:/FR"A
>R21:/-*-
>R20:"RIGHT
>E20:"HERE
>A20:"TTLV
>E14:/FL+E13*2
>D14:/FL
>A14:/FR"FOUR
>E13:/FL2
>A13:/FR"TWO
>A12:/-+-
>A11:"LEFT
/W1
/GOC
/GRA
/XV15
/GC4
/X>A1:>A1:/TV
/X-/X>C11:>E20:;/GC7
/X!/X>R20:>T29:;
```

FIGURE 4. *A printout of the sheet shown in Figure 5.*

Although we will briefly explain how some of the /X commands work in the particular sheet developed for this article, when you develop your own sheet, do not use any form of the /X command, as the details of their operation cannot be guaranteed in any way. In some instances they may reset column widths, in others they may reset titles.

The command /XV15 sets a vertical window beginning at the 15th character column, subject to the caveats noted above.

Commands beginning with /X>, in general, identify the upper left corner of a window. However, in files in which title commands appear, /X> identifies the beginning of a title boundary, as shown in the line /X>A1:>A1:/TV. In effect, a title boundary defines a window within a window, and /X> thus defines the upper left of the title "window."

The line beginning with /X– contains a number of commands. The first element, /X–, is a command that is specific to machines with two arrow keys (the Apple II, for example), though it can be loaded on other machines. It sets the direction indicator in the current window.

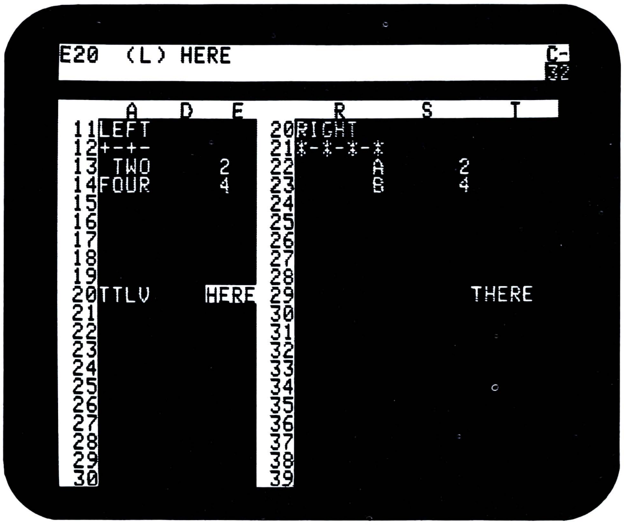
The remainder of that line, /X>C11:>E20:;/GC7, defines C11 as the upper left corner of the LEFT window; moves the cursor to E20; shifts the cursor to the RIGHT window with the semicolon; and, finally, sets the column width in the RIGHT window to 7.

Notice that columns B and C have been scrolled to the left and are not visible in the screen photo, though column A, which is the title boundary, is visible. The upper left of the LEFT window is defined as C11 so that columns D and E will appear in the LEFT window when the sheet is displayed.

The final line:

```
/X!/X>R20;>T29:;
```

is similar to the preceding one. First, it sets the direction indicator with /X!; defines the upper left of the RIGHT window at R20; positions the cursor at T29; then returns the cursor to the first window. Since the last position of the cursor in the first window was



```
E20  (L) HERE                              C-
                                           32
   A    D    E          R        S       T
11LEFT               20RIGHT
12+-+-               21*-*-*-*
13 TWO       2       22       A        2
14FOUR       4       23       B        4
15                   24
16                   25
17                   26
18                   27
19                   28
20TTLV     HERE      29                 THERE
21                   30
22                   31
23                   32
24                   33
25                   34
26                   35
27                   36
28                   37
29                   38
30                   39
```

FIGURE 5. *A screen photo of a saved sheet.*

E20, that is where it will appear when the display is complete.

Generally, if you are typing entries from a printout, you can use /X as shown on the printout without encountering difficulties. We recommend not normally using /X from the keyboard, since its behavior is not well defined and it can have some odd and unexpected effects on the screen display, though it will not harm the program or the data.

VISICALC POINTER . . .

On many machines, the letters VC appear in the upper left corner of the screen while the program is loading. Why?

The reason varies from machine to machine. VC serves to alert you to the fact that the program is being loaded into the computer's memory. In some cases the VC display is read by the program to ascertain the model of the computer into which it is being loaded. In the case of the TRS-80 Model I, for example, the display of VC enables the program to determine whether the computer has been modified to display both lower and upper case letters. Many early TRS-80s do not have that capability. On the Commodore machines, the act of displaying the VC allows the VisiCalc® program to determine if it is being loaded on a 40 column or an 80 column machine.

The VC message can also be informative. The Apple II VC-202B0-AP2 version displays VC16 to indicate that a 16 sector version of the VisiCalc program is being loaded, VC13 for 13 sector, and VCBP for 13 sector being loaded via the Apple BASICS diskette's boot program.

CARRYING ON WITH VISICALC

Letters written home by students often echo a theme dating from Medieval times: "Send more money." It's a plea that has been entered in many languages, and the purpose of this article is to do so once again, in the language of mathematics. Naturally, given the source and nature of this journal, we'll enter it on a VisiCalc® sheet.

The method we will use to solve the problem at hand is not the simplest of all possible methods, but it makes several important points about using the VisiCalc program. We'll show, for example, how to avoid creating a forward reference, and how to break down an expression that is too long, as well as give you some pointers on using /R. We also show how to use the VisiCalc program to write a modulo function using @INT.

The problem itself is to assign a different, one digit numerical value to each of the eight letters required to write the statement "SEND MORE MONEY," such that, when the values are substituted for the letters, SEND + MORE = MONEY. For example, although 5082 + 4960 = 10042, as shown in Figure 6 the value of the E in MONEY is 4, while in SEND and MORE it is 0, meaning that the values assigned to N and R are apparently incorrect.

Such puzzles are call cryptarithms, and follow three simple rules. First, the values assigned to each letter must be different, single digit integers; second, any given letter must always have the same numerical value, whether it appears in the addend or the sum; and, third, the values assigned must add correctly.

We should also make two observations about cryptarithms. First, a good one uses real words, not nonsense syllables. For example, WRITER + EDITOR = ARTICLE would be a good cryptarithm. CMXNER + BVDCMR = CDCXEBM would not. Second, though some cryptarithms may have two or three solutions, better ones have only one.

To an experienced puzzler, solving a cryptarithm is child's play. For most of the rest of us it takes more effort, especially if working with pencil and paper. It's a great deal easier to solve them with the VisiCalc program. Be forewarned: This article will show you how to use the VisiCalc program to solve cryptarithms more easily, but not how to actually solve them. That's why the "solution" in Figure 6 is incorrect.

Building a Cryptarithm on a VisiCalc Sheet

Begin by setting up your sheet with /GOR, to calculate across rows, and, since only two cells display more than a single digit or letter, with /GC3, to go to a column width of 3 characters. In row 1, enter each of the letters used in the cryptarithm in cells B1 . . . I1. Since M, O, N, and E are used more than once, we need only eight cells to accommodate the entire inventory of letters.

Directly below, in cells B2 . . . I2, are guesses at the values of the individual letters. It will be easier to follow our discussion if you enter the values shown in Figure 6.

Below and to the left of the guesses is the cryptarithm itself, which serves as a guide to the order in which we must enter the values and expressions that lay to the right. There are no values or expressions in these cells.

To create the first addend, we must bring the guesses in row 2 into row 6. Enter in cells I6 . . . L6:

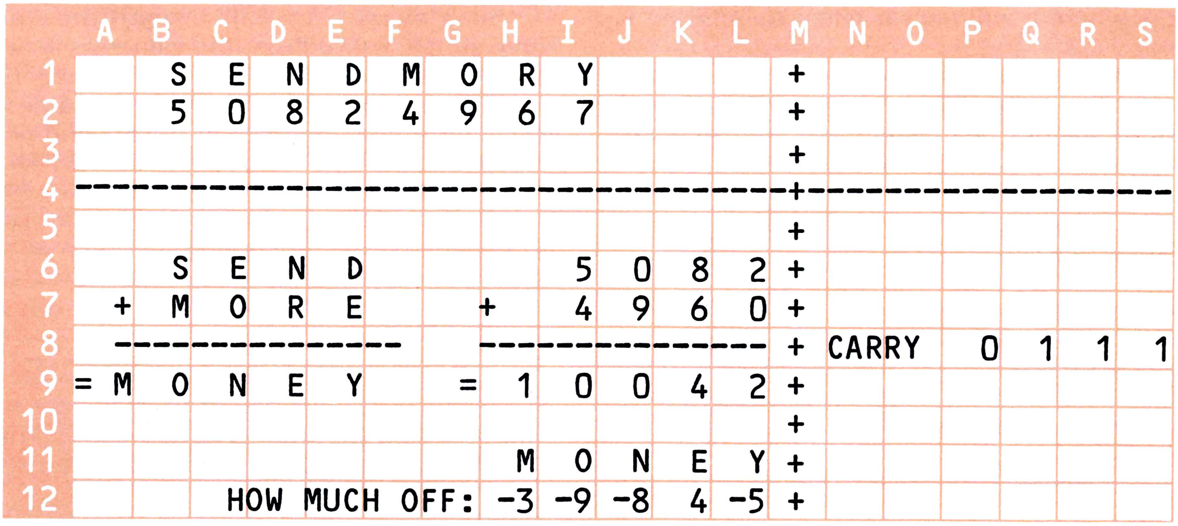
```
>I6:+B2
>I6:/R:J6 . . . L6:R
```

In the corresponding cells in row 7, create the second addend with the entries:

```
>I7:+F2
>J7:+G2
>K7:+H2
>L7:+C2
```

Because the entry at L7 is +C2, you cannot use /R as with the first addend. In that case, when you specified that the entries in the target range were relative, the VisiCalc program entered the contents of cells C2, D2, and E2 in J6, K6, and L6, which is to say that the program assumes that you are progressing in the same direction through the alphabet. If you specified relative references in the entries for the second addend, the entry at L7 would have been +I2, which is the value for Y, not D.

Stop here and save the sheet with /SS, giving it the file name CRITHM.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1		S	E	N	D	M	O	R	Y				+						
2		5	0	8	2	4	9	6	7				+						
3													+						
4	---	---	---	---	---	---	---	---	---	---	---	---	+	---	---	---	---	---	---
5													+						
6		S	E	N	D				5	0	8	2	+						
7	+	M	O	R	E			+	4	9	6	0	+						
8	---	---	---	---	---			---	---	---	---	---	+	CARRY		0	1	1	1
9	= M	O	N	E	Y		=	1	0	0	4	2	+						
10													+						
11								M	O	N	E	Y	+						
12			HOW MUCH OFF:					-3	-9	-8	4	-5	+						

FIGURE 6. *The layout of a cryptarithm sheet.*

Adding the Cryptarithm

The screen will still display the values and labels you entered before saving your sheet. If you now add the values that appear in rows 6 and 7 by entering:

```
>H9:+H6+H7
>H9:/R:I9 . . . L9:RR
```

you will get the erroneous and strange looking result:

```
    5  0  8  2
 +  4  9  6  0
 0  9  914  2
```

What has happened is that the VisiCalc program has performed the additions correctly, placing in the designated cells the sums of the two cells above.

But what is needed in cells H9 . . . L9 is a series of expressions that will add the values in the rows directly above; and, if the sum is equal to or greater than 10, deduct the 10; then add the number of 10's deducted to the values in the next column. In other words, we must make the program behave as we do when we "carry 10's" to add a column of figures.

To do that we will employ a mathematical function called modulo, or "mod," for short.

It's What's Left That Counts

While the term "modulo" may be unfamiliar, its use is so common that it is barely noticed. Each time we add two digits that total more than 9, we employ modulo 10 to place the digits in the sum in the correct sequence. Ordinarily, we would state a simple addition problem as follows:

8 + 6 = 14

However, we've seen that that is not what we want on the VisiCalc sheet we have developed. All that is needed there is the digit 4, so we can restate the problem as:

(8 + 6) modulo 10 is 4

Modulo, then, yields the remainder, the value of the digit that is placed directly below the digits being added. In essence, the term modulo tells us that we are to subtract some number (the modulus) from another until we can no longer do so without generating a negative number.

Mod has a variety of uses beyond playing mathematical games. It is often applied by intelligence agencies in the creation of ciphers. As we'll see in a future issue of SATN, mod can also be useful in business to generate part numbers that are concise and compact, and still contain large amounts of information.

Creating Mod with @INT

Because the problem that is the subject of this article is one of adding two numbers expressed in a decimal system, the only modulus used is 10. In other numbering systems, other values are assigned to mod. In a binary (base 2) system, for example, the modulus is 2, and in a hexadecimal (base 16) system, 16.

The general formula to create mod 10 with the VisiCalc program is:

*Value – (@INT(value/10) * 10)*

The value in that formula can be either a single entered value or, as on the sheet in our example, an expression like +L6+L7. By substituting 2 or 16 for 10 in the formula we would have a formula for the mods required in binary or hexadecimal systems.

Now enter:

```
>L9:+L6+L7-(@INT((L6+L7)/10)*10)
```

If you have been entering the values and expressions shown in Figure 6, the value displayed at L9 will not have changed. However, if you now go to C2 and change the value assigned to E to 8 or 9, you'll see that 0 or 1 is returned at L9. Or, you could go to E2 and change the value assigned to D. Whichever value you change, the value returned at L9 will always be:

+L6 + L7 modulo 10

But consider the implications of using the expression we have designed for mod when adding the values assigned to N and R. At K9 we need the sum of the values at K6 and K7, plus 1 if +L6+L7 total more than 10, or 0 if they total 10 or less. We could insert the expression at L9 into the one at K9, but this is the expression that would result:

```
>K9:+K6+K7+@INT((L6+L7)/10)-(@INT
 ((K6+K7+@INT((L6+L7)/10))/10)*10)
```

There is nothing mathematically wrong with that expression; it will yield exactly the result needed at K9. But it is unwieldy and difficult to manage. Such expressions are best broken down whenever possible. That's the purpose of the CARRY field to the right of the values we are adding.

Now clear the screen with /CY and load the file CRITHM.

Carry On

To the right of the label CARRY are portions of the formula for mod. The entries in the field calculate the number of 10's that must be carried from column to column as the values representing the letters in SEND and MORE are added. The portion of the mod formula that we use is that which gives the integer of the expression:

Value/10

Or, for the first entry in the CARRY field:

```
>P8:@INT((L6+L7)/10)
```

Compare the entry at P8 with the expression for mod developed above:

```
+L6+L7-(@INT((L6+L7)/10)*10)
```

And you'll see that what remains to complete the addition of the values representing D and E is to multiply P8 by 10 and subtract the result from +L6+L7. That is done at L9 with the entry:

```
>L9:+L6+L7-(P8*10)
```

Substituting any two single digit values for those currently at C2 and E2 will return the correct values at L9 and P8. Zero and one, for example, will return 1 at L9 and 0 at P8; 6 and 7 will return 3 at L9 and 1 at P8.

Now make the following entries to fill out the CARRY field:

```
>Q8:@INT((K6+K7+P8)/10)
>R8:@INT((J6+J7+Q8)/10)
>S8:@INT((I6+I7+R8)/10)
```

Notice that the alphabetical progression within the entries in the CARRY field is P, Q, R, S, and L, K, J, I. That means that to evaluate the expressions, the VisiCalc program moves both to the left and right in the alphabet. It also means, as above, that you cannot replicate the entry in Q8 into R8 . . . S8.

At first glance the appearance of the CARRY field may seem awkward, but there is no other way to set up the sheet so that it can recalculate automatically and across rows and yield the correct results immediately. The structure of the entries could be altered, but that would mean that each time a value was changed in row 2, ! would have to be used to get the correct result.

Notice, too, that the CARRY field can be placed only on rows 7 or 8, since the entries in the field must be evaluated before the entries in row 9 can be evaluated correctly. Placing the CARRY field in any row except 7 or 8 creates a forward reference that would necessitate recalculating with ! if you wish to see the correct result of changing any value in row 2.

Summing Up

Now, adding the values assigned to the letters in SEND and MORE is a matter of taking the sum of the values in the appropriate cells above and in the CARRY field and subtracting the appropriate amount to carry. Enter:

```
>K9:+K6+K7+P8-(Q8*10)
>J9:+J6+J7+Q8-(R8*10)
>I9:+I6+I7+R8-(S8*10)
>H9:+S8
```

At this point you can return to row 2 and begin substituting values to see if you can make SEND + MORE = MONEY. To track how closely your guesses come to the actual values required, however, it's best to add two more elements shown on the sample sheet: the word MONEY, which serves as another guide; and the field HOW MUCH OFF:, which contains the entries:

```
>H12:+H9-F2
>I12:+I9-G2
>J12:+J9-D2
>K12:+K9-C2
>L12:+L9-I2
```

Then, as you change the values in row 2, the signs and sizes of values in row 18 will tell you how near or far from the mark your guesses are. For example, if M=4, as shown in Figure 6, then HOW MUCH OFF: shows that we have guessed too high by 3. Changing F2 to 0, though, has an effect all through the cryptarithm.

It's worth adding that cryptarithms have been used as more than mere entertainment. How different people solve them has provided cognitive psychologists with insights into human problem solving abilities. And how cryptarithms are solved by machines has provided computer scientists with useful information on artificial intelligence. An interesting look at both those topics can be found in the book, *Human Problem Solving* by Allen Newell and Herbert Simon (Englewood Cliffs: Prentice Hall, 1972).

Now you are on your own. We could give you an algorithm that would make working out the correct values relatively simple, but we would like to think that you would prefer to develop one of your own. Of course, there's the one algorithm that works eventually for every problem: trial and error. There is one correct solution to SEND + MORE = MONEY, but if you don't get it immediately, don't worry unduly; we'll publish it in around eight weeks. Depending on how hard you work at it, by then you may have run through all 1,814,400 possibilities.

WHO'S IN FIRST?

Part of the fun of following sports is playing with the numbers. That's recognized by newspapers where, every day, as much as a full page may be devoted to box scores, player and team statistics, league standings, and so forth. Given the space available, we can't begin to cover the hundreds of measurements that are made for all sports, but we have been intrigued by the problem of answering this question: "How do you figure out how many games ahead a given team is?" (One of the more mathematically inclined programmers here at Software Arts claimed that the answer was derived by magic.)

Besides the "fun" aspect of such calculations, there is also a practical one. If you are involved in amateur athletics, high school sports, for example, keeping track of standings is one of the things that the VisiCalc® program can help with.

Both baseball and basketball standings are calculated in the same way. At the professional level, tie games are not permitted in either sport except in the rarest of circumstances. Since the basketball season had not begun as we wrote this, our example will show only baseball teams.

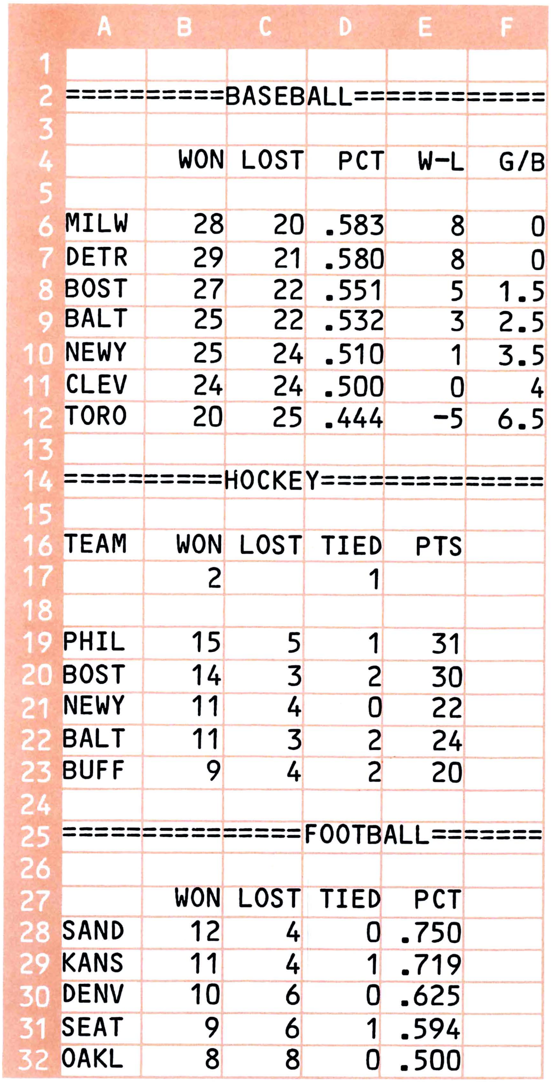
Baseball Standings

Let's begin by setting up a VisiCalc sheet as shown in Figure 7. In the first row enter, in cells B1 . . . F1, the labels WON, LOST, PCT (for percentage), W-L (for wins minus losses), and G/B (for games behind). In column A, beginning at A6, enter the teams in the league or division. In our example we show the teams in the American League East. The sheet should also be set to calculate down columns with /GOC, which is the default condition if you have just booted the program or used /CY.

Under WON and LOST we have entered the actual records of the teams in the AL East as of October 1, a few days before the split season ended.

The teams' winning percentages are calculated in column D by the general formula:

Games Won / Games Played



	A	B	C	D	E	F
1						
2	==========BASEBALL===========					
3						
4		WON	LOST	PCT	W-L	G/B
5						
6	MILW	28	20	.583	8	0
7	DETR	29	21	.580	8	0
8	BOST	27	22	.551	5	1.5
9	BALT	25	22	.532	3	2.5
10	NEWY	25	24	.510	1	3.5
11	CLEV	24	24	.500	0	4
12	TORO	20	25	.444	-5	6.5
13						
14	==========HOCKEY=============					
15						
16	TEAM	WON	LOST	TIED	PTS	
17		2		1		
18						
19	PHIL	15	5	1	31	
20	BOST	14	3	2	30	
21	NEWY	11	4	0	22	
22	BALT	11	3	2	24	
23	BUFF	9	4	2	20	
24						
25	===============FOOTBALL=======					
26						
27		WON	LOST	TIED	PCT	
28	SAND	12	4	0	.750	
29	KANS	11	4	1	.719	
30	DENV	10	6	0	.625	
31	SEAT	9	6	1	.594	
32	OAKL	8	8	0	.500	

Figure 7. *A sheet for calculating team standings. Note that the sheet shown is set up with /GC5.*

Which is calculated on our sheet by the entry:

```
>D6:+B6/(@SUM(B6 . . . C6))+.00000001
```

The reason for adding .00000001 is so that the screen will display each team's PCT to three digits, while still not significantly affecting the actual percentage standings. That is, if a team is playing .500 ball, the VisiCalc program will display .5, but the addition of a very small increment will force a display of .500. (You cannot use this ploy if your calculations require accuracy to eight places or more, but that's not the case in baseball.)

The expression at D6 is then replicated relatively into the cells for the rest of the teams:

```
>D6:/R:D7 . . . D12:RRR
```

Any team's position in its division depends on the other teams' net wins, so we have included the column W-L to help determine that. W-L is simply the number of games above or below .500, and we calculate that for all the teams in the division with the expression:

```
>E6:+B6-C6
```

VISICALC POINTER . . .

When you enter a number on a large sheet, the VisiCalc® program seems to take a long time to finish the recalculation.

Normally, all of the values on a sheet are recalculated each time one value is changed. This "automatic recalculation" means that if there are many values to be recalculated, or complex expressions to be evaluated, it may take a long time. To save time, you can set the program to "manual recalculation mode" by using /GRM (Global Recalculate Manual). Thereafter, you must use the ! command to request recalculation, but /GRM is very useful when you want to change several values before seeing the final results. To return to the automatic recalculation mode, use the /GRA command.

Which we replicate as:

```
>E6:/R:E7 . . . E12:RR
```

In effect, any baseball game is a contest for a single point in the standings. However, that point is split by the two teams in the game, with the winner gaining a positive half point (+.5) and the loser gaining a negative half point (−.5). Thus, the positive half point gained by any team when it wins a game is cancelled by the negative half point it gains when it loses a game, and the W-L column becomes all important in determining standings.

To calculate the actual standings, then, we must compare each team's W-L to the W-L of the team with the best record and take the difference of the two. Once we've got that difference, we must multiply it by .5. We use .5 because we have already accounted for the teams' negative half points by calculating the W-L column. It also allows us to express the number of games behind as a positive figure, though in fact it is a negative.

To make the comparison we use the @MAX function of the VisiCalc program. Enter:

```
>F6:(@MAX(E6 . . . E12)-E6)*.5
```

Now we have to replicate that expression down the sheet. Since we are comparing each team's record to the record of the team with the largest number of games over .500, keep the @MAX term of the expression the same. That is, replicate it with no change. We are concerned with each team's relative number of games over .500, though, so that part of the expression must be relative. Enter:

```
>F6:/R:F7 . . . F12:NNR
```

As of the morning of October 1st, Milwaukee had won 28 and lost 20, so its W-L was 8. So was Detroit's, though it had won 29, and lost 21. Thus, the teams were tied for first place in terms of games, but Milwaukee led the division by 3 "percentage points" (.3 percent, actually).

Hockey and Football Standings

In both hockey and football, ties are an integral part of the standings, though they are figured slightly differently in each case. We'll look first at hockey, which is the more straightforward calculation.

As in baseball and basketball, each hockey game is a contest for points. However, there are no half points involved. The winning team gains two points, the losing team none. Ties are common, though, and when they occur each of the two teams gains a single point. Because losses generate no points, it's as though the game did not exist for the team that lost.

Thus, the VisiCalc sheet is simple to construct and the expression to calculate hockey standings is a matter of multiplying each team's point total for its games won and tied, and adding the results.

As on the sheet for baseball standings, we have entered teams in column A, beginning at A19. In B16 . . . E16 are the labels WON, LOST, TIED, and PTS. Beneath the labels WON and TIED are entries for the number of points awarded for each. Since no points are awarded for a loss, there is no need to enter any point value. Thus, enter:

```
>B17:2
>D17:1
```

At E19, the PTS column for the first team, is the expression that calculates that team's point total. That is:

```
>E19:(B19*B17)+(D19*D17)
```

Now replicate that into the cells for the other teams, keeping their respective wins and ties relative, and the points awarded for wins and ties unchanged. Enter:

```
>E19:/R:E20 . . . E23:RNRN
```

If you wish, you can add a column to those shown to measure how many points behind a team is. The expression to do so is identical to the expression used to calculate a baseball or basketball team's number of games behind, except for the final term (*.5), which is left off.

In football, ties are treated as half games won. That is, a team's percentage standing is calculated by the general formula:

*Wins + (Ties * .5) / Games Played = Pct*

After setting up the VisiCalc sheet as shown in Figure 7 and entering the teams and their hypothetical records (with apologies to Raiders' fans), we have entered at E28 the expression:

```
>E28:(+B28+(D28*.5))/(@SUM
 (B28 . . . D28))+.00000001
```

As above, the final term was added to display three digits in the PCT column. (If a team is winless, the .00000001 will display as 1E-7, and if you would rather show a 0 PCT, you may eliminate the .00000001 from the expressions.)

That expression is replicated relatively into the remaining cells for the other teams in the division as follows:

```
>E28:/R:E29 . . . E32:RRRR
```

Now, at least, you can keep track of "who's in first," using the VisiCalc program. We can't promise at this time that we'll ever be able to show you how to use the program to predict who will win the World Series, but if we find out how the little boy in the TV commercial does it, we'll pass it on.

DON'T MISS AN ISSUE!

If you haven't already subscribed to SATN, now is the time to do it. A subscription costs $30 for 6 issues ($50 outside USA) — just send a check or money order, payable to Software Arts, Inc., to: SATN Subscriptions, P.O. Box 815, Quincy, MA 02169.